Express Label: EL140119624US
Deposited: November 4, 1998

REDUCED-ERROR PROCESSING OF TRANSFORMED DIGITAL DATA

CROSS REFERENCES

The present application is related to the following applications even dated herewith: Attorney docket number YO998-372, entitled, "Transform-domain correction of real-domain errors," by inventors J. Mitchell et al., and Attorney docket number YO998-373, entitled, "Error reduction in transformed digital data," by inventors M. Bright et al., which are incorporated herein in entirety by reference.

FIELD OF THE INVENTION

This invention relates to transform coding of digital data, specifically to real domain processing of transform data. More particularly, this invention relates to reduced-error digital processing of inverse transformed data.

BACKGROUND OF THE INVENTION

Transform coding is the name given to a wide family of techniques for data coding, in which each block of data to be coded is transformed by some mathematical function prior to further processing. A block of data may be a part of a data object being coded, or may be the entire object. The

data generally represent some phenomenon, which may be for example a spectral or spectrum analysis, an image, an audio clip, a video clip, etc. The transform function is usually chosen to reflect some quality of the phenomenon being coded; for example, in coding of audio, still images and motion pictures, the Fourier transform or Discrete Cosine Transform (DCT) can be used to analyze the data into frequency terms or coefficients. Given the phenomenon being coded, there is generally a concentration of the information into a few frequency coefficients. Therefore, the transformed data can often be more economically encoded or compressed than the original data. This means that transform coding can be used to compress certain types of data to minimize storage space or transmission time over a communication link.

An example of transform coding in use is found in the Joint Photographic Experts Group (JPEG) international standard for still image compression, as defined by *ITU-T Rec. T.81 (1992) | ISO/IEC 10918-1:1994, Information technology — Digital compression and coding of continuous-tone still images, Part 1: Requirements and Guidelines*. Another example is the Moving Pictures Experts Group (MPEG) international standard for motion picture compression, defined by ISO/IEC 11172:1993, *Information Technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbits/s*. This MPEG-1 standard defines systems for both video compression (Part 2 of the standard) and audio compression (Part 3). A more recent MPEG video standard (MPEG-2) is defined by *ITU-T Rec. H.262 | ISO/IEC 13818-2: 1996 Information Technology —*

Generic *Coding of moving pictures and associated audio -- Part 2: video.* A newer audio standard is *ISO/IEC 13818-3: 1996 Information Technology* — Generic *Coding of moving pictures and associated audio -- Part 3: audio.* All three image international data compression standards use the DCT on 8x8 blocks of samples to achieve image compression. DCT compression of images is used herein to give illustrations of the general concepts put forward below; a complete explanation can be found in Chapter 4 "The Discrete Cosine Transform (DCT)" in W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard,* Van Nostrand Reinhold: New York, (1993).

Wavelet coding is another form of transform coding. Special localized basis functions allow wavelet coding to preserve edges and small details. For compression the transformed data is usually quantized. Wavelet coding is used for fingerprint identification by the FBI. Wavelet coding is a subset of the more general subband coding technique. Subband coding uses filter banks to decompose the data into particular bands. Compression is achieved by quantizing the lower frequency bands more finely than the higher frequency bands while sampling the lower frequency bands more coarsely than the higher frequency bands. A summary of wavelet, DCT, and other transform coding is given in Chapter 5 "Compression Algorithms for Diffuse Data" in Roy Hoffman, *Data Compression in Digital Systems*, Chapman and Hall: New York, (1997).

In any technology and for any phenomenon represented by digital data, the data before a transformation is performed are referred to as being "in the real domain". After a

transformation is performed, the new data are often called "transform data" or "transform coefficients", and referred to as being "in the transform domain". The function used to take data from the real domain to the transform domain is called the "forward transform". The mathematical inverse of the forward transform, which takes data from the transform domain to the real domain, is called the respective "inverse transform".

In general, the forward transform will produce real-valued data, not necessarily integers. To achieve data compression, the transform coefficients are converted to integers by the process of quantization. Suppose that (λ_i) is a set of real-valued transform coefficients resulting from the forward transform of one unit of data. Note that one unit of data may be a one-dimensional or two-dimensional block of data samples or even the entire data. The "quantization values" (q_i) are parameters to the encoding process. The "quantized transform coefficients" or "transform-coded data" are the sequence of values (a_i) defined by the quantization function Q:

$$a_i = Q(\lambda_i) = \left\lfloor \frac{\lambda_i}{q_i} + 0.5 \right\rfloor \qquad (1)$$

where $\lfloor x \rfloor$ means the greatest integer less than or equal to x. The resulting integers are then passed on for possible further encoding or compression before being stored or transmitted. To decode the data, the quantized coefficients are multiplied by the quantization values to give new "dequantized coefficients" (λ_i') given by

$$\lambda_i' = q_i a_i . \qquad (2)$$

The process of quantization followed by dequantization (also called inverse quantization) can thus be described as "rounding to the nearest multiple of q_i". The quantization values are chosen so that the loss of information in the quantization step is within some specified bound. For example, for audio or image data, one quantization level is usually the smallest change in data that can be perceived. It is quantization that allows transform coding to achieve good data compression ratios. A good choice of transform allows quantization values to be chosen which will significantly cut down the amount of data to be encoded. For example, the DCT is chosen for image compression because the frequency components which result produce almost independent responses from the human visual system. This means that the coefficients relating to those components to which the visual system is less sensitive, namely the high-frequency components, may be quantized using large quantization values without perceptible loss of image quality. Coefficients relating to components to which the visual system is more sensitive, namely the low-frequency components, are quantized using smaller quantization values.

The inverse transform also generally produces non-integer data. Usually the decoded data are required to be in integer form. For example, systems for the playback of audio data or the display of image data generally accept input in the form of integers. For this reason, a transform decoder generally includes a step that converts the non-integer data from the inverse transform to integer data, either by truncation or by rounding to the nearest integer. There is also often a limit on the range of the integer data

output from the decoding process in order that the data may be stored in a given number of bits. For this reason the decoder also often includes a "clipping" stage that ensures that the output data are in an acceptable range. If the acceptable range is [*a*,*b*], then all values less than ***a*** are changed to ***a***, and all values greater than ***b*** are changed to ***b***.

These rounding and clipping processes are often considered an integral part of the decoder, and it is these which are the cause of inaccuracies in decoded data and in particular when decoded data are re-encoded. For example, the JPEG standard (Part 1) specifies that a source image sample is defined as an integer with precision P bits, with any value in the range 0 to *2***P - 1. The decoder is expected to reconstruct the output from the inverse discrete cosine transform (IDCT) to the specified precision. For the baseline JPEG coding P is defined to be 8; for other DCT-based coding P can be 8 or 12. The MPEG-2 video standard states in Annex A (Discrete cosine transform) "The input to the forward transform and the output from the inverse transform is represented with 9 bits."

For JPEG the compliance test data for the encoder source image test data and the decoder reference test data are 8 bit/sample integers. Even though rounding to integers is typical, some programming languages convert from floating point to integers by truncation. Implementations in software that accept this conversion to integers by truncation introduce larger errors into the real-domain integer output from the inverse transform.

The term "high-precision" is used herein to refer to numerical values which are stored to a precision more accurate than the precision used when storing the values as integers. Examples of high-precision numbers are floating-point or fixed-point representations of numbers.

SUMMARY OF THE INVENTION

In light of the problems described above regarding inaccuracies caused by digital processing techniques and by such things as rounding and clipping after the inverse transform of transform data, one aspect of this invention provides a method for processing transform data in the real domain. This method reduces the undesired errors in the data produced by such things as rounding to integers and clipping to an allowed range after the inverse transform. In an embodiment, this method includes: performing the inverse transform of the transform data such that the real-domain data produced are in the form of high-precision numbers; processing these high-precision numbers; and converting the processed high-precision numbers to integers and clipping to an allowed range only after the processing stage is complete.

It is another aspect of this invention to provide a method for processing transform-coded data in the real domain which reduces the undesired errors in the data produced by the converting to integers and clipping to an allowed range after the inverse transform. In an embodiment, the method includes: performing the inverse

quantization of the transform-coded data; performing the inverse transform of the transform data thus produced, such that the real-domain data produced are in the form of high-precision numbers; processing these high-precision numbers; and converting the processed high-precision numbers to integers and clipping to an allowed range only after the processing stage is complete.

Still another aspect of the present invention is to provide a method for processing transform-coded data in the real domain to produce new transform-coded data, which reduces the error produced by converting to integers and clipping to an allowed range after the inverse transform. In an embodiment, this method includes: performing the inverse quantization of the transform-coded data; performing the inverse transform of the transform data thus produced, such that the real-domain data produced are in the form of high-precision numbers; processing these high-precision numbers; performing the forward transform on the processed high-precision numbers; and performing quantization on the new transform data. If the errors in the forward and inverse transforms and in the processing are sufficiently small, there will be no undesirable errors produced in the new quantized transform-domain data.

There is no requirement that the input data to the methods described herein need come from a single data source. Thus, this invention is not restricted to the real-domain processing of data from a single source, but also applies to real-domain processing of data from multiple sources, such as the merging of images or audio data.

The quantization described in the background is the linear quantization used in international image data compression standards such as JPEG and MPEG. There is no requirement that the quantization be linear. Any mapping that reduces the number of transform data levels in a deterministic way can be used with this invention. The quantization step has been described mathematically with a division in Equation (1). Actual embodiments may use a lookup table or a sequence of comparisons to achieve similar results.

It is a further aspect of the invention to provide apparatus, a computer product and an article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for causing a computer to perform the methods of the present invention.

BRIEF DESCRIPTION OF FIGURES

These and other objects, features, and advantages of the present invention will become apparent upon further consideration of the following detailed description of the invention when read in conjunction with the drawing figures, in which:

FIG. 1(a) is a block diagram showing a method for performing an inverse transform;

FIG. 1(b) is a block diagram showing a system for performing an inverse transform;

FIG. 2(a) is a block diagram showing a method for decoding transform-coded data;

FIG. 2(b) is a block diagram showing a system for decoding transform-coded data;

FIG. 3 is a block diagram showing a method for the real-domain processing of transform data;

FIG. 4 is a block diagram showing a method for performing an inverse transform followed by a forward transform, and demonstrating the multi-generation problem;

FIG. 5 is a block diagram showing a method for decoding and re-encoding transform-coded data, and demonstrating the multi-generation problem;

FIG. 6 is a block diagram showing a method for performing an inverse transform, real-domain data manipulation and a forward transform, and demonstrating the multi-generation problem;

FIG. 7(a) is a block diagram showing a method for performing real-domain processing of JPEG DCT-coded image data, which exhibits the multi-generation problem;

FIG. 7(b) is a block diagram showing a system for performing real-domain processing of JPEG DCT-coded image data, which exhibits the multi-generation problem;

Fig. 8(a) gives the JPEG example luminance quantization matrix;

Fig. 8(b) gives the JPEG example chrominance quantization matrix;

FIG. 8(c) is a numerical example of how real-domain rounding can cause significant errors in 8x8 block DCT coded data;

FIG. 8(d) is a numerical example of how real-domain truncation can cause significant errors in 8x8 block DCT coded data;

FIG. 8(e) is a series of graphs illustrating how real-domain clipping can cause errors in one-dimensional discrete cosine transform-coded data;

FIG. 8(f) and FIG. 8(g) are a numerical example of how real-domain clipping can cause significant errors in 8x8 block DCT coded data;

FIG. 9 is a block diagram showing a method performing multiple iterations of the process described in FIG. 5, and exhibiting the multi-generation problem;

FIG. 10 is a block diagram showing a method for performing multiple iterations of real-domain manipulations, and exhibiting the multi-generation problem;

FIG. 11(a) is a block diagram showing an example of a method for reduced-error processing of transform data in accordance with the present invention;

FIG. 11(b) is a block diagram showing an example of a system for reduced-error processing of transform data in accordance with the present invention;

FIG. 12(a) is a block diagram showing an example of a method for performing an inverse transform followed by a forward transform, such that this process is lossless in accordance with the present invention;

FIG. 12(b) is a block diagram showing an example of a system for performing an inverse transform followed by a

forward transform, such that this process is lossless in accordance with the present invention;

FIG. 13(a) is a block diagram showing an example of a method for performing real-domain manipulation of transform data with reduced error followed by a forward transform in accordance with the present invention;

FIG. 13(b) is a block diagram showing an example of a system for performing real-domain manipulation of transform data with reduced error followed by a forward transform in accordance with the present invention;

FIG. 14(a) is a block diagram showing an example of a method for reduced-error processing of transform-coded data in accordance with the present invention;

FIG. 14(b) is a block diagram showing an example of a system for reduced-error processing of transform-coded data in accordance with the present invention;

FIG. 15(a) is a block diagram showing an example of a method for decoding and re-encoding transform-coded data such that this process is lossless in accordance with the present invention;

FIG. 15(b) is a block diagram showing an example of a system for decoding and re-encoding transform-coded data such that this process is lossless in accordance with the present invention;

FIG. 16(a) is a block diagram showing an example of a method for performing real-domain manipulation of transform-coded data with reduced error in accordance with the present invention;

FIG. 16(b) is a block diagram showing an example of a system for performing real-domain manipulation of transform-coded data with reduced error in accordance with the present invention;

FIG. 17(a) is a block diagram showing an example embodiment of a method for performing real-domain processing of JPEG-coded image data, such that undesired errors in the new transform-coded data are reduced or eliminated in accordance with the present invention;

FIG. 17(b) is a block diagram showing an example embodiment of a system for performing real-domain processing of JPEG-coded image data, such that undesired errors in the new transform-coded data are reduced or eliminated in accordance with the present invention;

FIG. 18(a) is a block diagram showing an example of a method for performing multiple iterations of the real-domain manipulation of transform-coded data with reduced error, where each iteration is as described in FIG. 16(a) in accordance with the present invention;

FIG. 18(b) is a block diagram showing an example of a system for performing multiple iterations of the real-domain manipulation of transform-coded data with reduced error, where each iteration is as described in FIG. 16(b) in accordance with the present invention;

FIG. 19(a) shows the same 8x8 block numerical starting point of FIG. 8(c) using the high-precision numbers as input to the forward transform instead of the rounded numbers;

FIG. 19(b) shows the same 8x8 block numerical starting point of FIG. 8(d) using the high-precision numbers as input to the forward transform instead of the truncated numbers;

FIG. 19(c) shows the same 8x8 block numerical steps as FIG. 8(f); and

FIG. 19(d) shows the numerical results when the output of the inverse DCT with rounding, but before clipping, is input to the forward transform followed by quantization.

DESCRIPTION OF THE PROBLEM

This invention provides methods, systems, and computer products which reduce or eliminate errors introduced by the processing of digital data. Firstly, the source of the error is analyzed and described. This is followed by a presentation of the invention concepts for error reduction and elimination. It is particularly noted that data manipulation and/or processing as employed here-to-before used digital techniques contaminated by the continued introducing of errors by the respective implementation of digital processing. These techniques employed for years are responsible for an inability to maintain original data precision and the continued deterioration of the data representing the phenomenon as more processing is performed. This is particularly detrimental when a process is performed on data which contain errors imparted on the data by previous processes. This results in the continued impairment of the data which thereby becomes less and less useful as more and more processes are performed thereupon.

The seriousness of the problem as realized by the inventors of the present invention is described forthwith. It is noted that in the figures presented herein, optional steps are often shown with dashed lines and/or boxes.

It is noted that the concepts of the present invention are useful in almost any digital processing technology. However, the subsequent description is mostly related to image data. This is because of the general availability and continued usage of image data compression standards which are employed worldwide. These standards require the introduction into the digital data of the errors to be described and the continued employment and processing of the error contaminated data. These standards basically teach away from the present invention. Thus image technology is a good example for describing the present invention.

Figure 1(a) shows an inverse transform method **100**. Transform-domain data 'A' **110** are acted on by the inverse transform **120**, which produces high-precision real-valued data **130**. The high-precision data **130** are converted to integers and clipped **140** to produce integer real-domain data **150**. In some cases, the integer-valued data are optionally sent to an output device **160**.

Figure 1(b) shows an inverse transform system **105**. Transform-domain data 'A' **115** are acted on by the inverse transformer **125**, which produces high-precision real-valued data **135**. The high-precision data **135** are input to the integer converter and clipper **145** to produce integer real-domain data **155**. In some cases, the integer-valued

data are optionally input to an output device **165** such as a display monitor, a television set, or an audio player.

Figure 2(a) shows a method **200** for decoding transform-coded (i.e. quantized) data. The integer transform-coded data 'B' **210** are inverse quantized **220** (i.e. dequantized) with quantization values as in Equation (2) above. The result of the dequantizing step may then be passed as input to the inverse transform **120**, and decoding proceeds as in Figure 1(a).

Figure 2(b) shows a system **205** for decoding transform-coded (i.e. quantized) data. The integer transform-coded data 'B' **215** are input to the inverse quantizer **225** with quantization values as in Equation (2) above. The result of the dequantizing step is passed as input to the inverse transformer **125**, and decoding proceeds as in Figure 1(b).

One aspect of the present invention is concerned with the manipulation of both transform data and transform-coded data. The words "manipulation" and "processing" are used interchangeably herein. Manipulation may be employed in order to achieve many different results. For example, image data must often be processed before printing by scaling and/or rotation. Data from two sources can be merged as is performed in chroma-keying of images or mixing of audio data. Manual manipulation of data is often needed for editing or color correction. Such manipulation of transform data are often performed on the integer real-domain data which results from the transform decoding of Figure 1(a) and/or Figure 2(a).

A process for manipulation of transform data **300** is shown in Figure 3. Integer data **150** undergo some form of manipulation **310**. If this manipulation **310** does not produce integer output, the manipulated output **340** is again converted to integers and clipped **320**. The resulting integer data **330** may be stored, transmitted, and/or optionally sent to an output device **160**. Because the stage of clipping and converting to integers **140** is performed before the manipulation which accepts integer input **150**, the resulting errors cause the data output from the manipulation **340** to contain at least small inaccuracies.

It is noted that there is no requirement in the data manipulation processes described above, for the input data to come entirely from one source. For example, many types of data manipulation involve the merging of data from two or more sources. This includes manipulations such as mixing of audio data or merging of images. The processes illustrated in the figures and described generally apply equally well to such types of manipulation. Thus the "input data" used for any of the processes described may in practice come from more than one input source.

It is often the case that data after manipulation are to be re-encoded to the transform domain. It is desirable that the process of decoding and re-encoding, when no manipulation is performed on the real-domain data, should be lossless. That is, the data, when the forward transform operation uses the same transform type operation as the inverse transform type of transform operation, should result in exactly the same transform-domain data as was present

initially. However, errors are introduced by the converting to integers and clipping to the allowed range as is illustrated in Figure 4. Figure 4 shows the integer data **150** used as input to the forward transform device **410**, which accepts integer-valued data as input. The resulting transform data 'A1' **420** are different from the original transform data 'A' **110** which were the input to the inverse transform. This is because the conversion to integers and the clipping process **140** have introduced errors into the process. The problem caused by the changes in data after each iteration, or "generation", of this process is herein called the "multi-generation problem".

The multi-generation problem is also illustrated for transform-coded data in Figure 5. Here the new transform-domain data **420** are quantized **510** to produce new transform-coded data 'B1' **520**. It is important to realize that the quantized data can only change if the errors produced are larger than half a quantization step:

$$Q(\lambda_i + \varepsilon) = Q(\lambda_i) \quad \text{if } |\varepsilon| < 0.5 q_i \qquad (3)$$

where ε is the error produced in this transform coefficient. This is because each of the λ_i is already a multiple of the quantization value, since they have been produced by dequantization as in Equation (2). Thus it is advantageous to control the errors so that they are sufficiently small. When the errors are sufficiently small, the new transform-coded data will be exactly the same as the original transform-coded data. The maximum possible error introduced by the conversion to integers by rounding is half the error introduced by truncating during the conversion.

Figure 6 shows a case wherein image manipulation is performed on the data and the resulting modified data are then re-transformed back to the transform domain. The integer data **150** are manipulated as was shown in Figure 3 to produce new integer-valued data **610**. These new integer-valued data **610** are used as the input to the forward transform **410** to produce new transform data 'A2' **620**. The fact that the process described above without any manipulation produces changes in the transform data **110** shows that when manipulation is performed there are undesired changes in the transform data **110** in addition to those which result from the desired manipulation.

An example of a method which embodies the process shown in Figure 6, is shown in Figure 7(a). The method **700** illustrated performs real-domain manipulation on coded data such as JPEG-coded image data. The coded data 'C' **710** are entropy decoded **720**, which is defined for JPEG-coded data in the JPEG standard. The entropy decode step **720** decompresses the data into quantized DCT coefficients. These quantized coefficients are inverse quantized **730** and passed to the inverse transform, which in this system is the two-dimensional 8x8 inverse DCT **740**. The resulting real-valued image data are rounded to integers and clipped **750** to the allowed range (e.g. [0,255]) to produce integer-valued image data **754** in the allowed range.

If it is necessary to show the data before manipulation, for example when the image manipulation is an interactive process, the image can optionally be sent to a display device **758**. The image is then manipulated **762** to

produce some desired change. If the result of the manipulation is non-integer data then the image data may be converted to integers and clipped to the range e.g. [0,255] **768**. In this way the image data **772** may again be displayed **758**. The new real-domain image data **772** are passed to the forward DCT **776** and the resulting DCT coefficients are quantized **780** to produce new quantized DCT coefficients **784**. These coefficients **784** are then entropy encoded **788** to produce new coded data 'C1' **792** which are different from the original coded data 'C' **710**. Now the new coded data 'C1' **792** incorporates not only the desired changes made to the image by the image manipulation **762**, but also the errors resulting from the converting and clipping stages **750** and **768**. It would be advantageous to eliminate or reduce these errors.

An example of a system which embodies the process shown in Figure 6, is shown in Figure 7(b). The system **705** performs real-domain manipulation on coded data. The coded data 'C' **715** are input to the entropy decoder **725**, which is defined for JPEG-coded data in the JPEG standard. The entropy decoder **725** decompresses the data into quantized DCT coefficients. These quantized coefficients are input to the inverse quantizer **735** and the output passed to the inverse transformer, which in this system is the two-dimensional 8x8 inverse DCT-er **745**. The resulting real-valued image data are rounded to integers and clipped **755** (e.g. to the range [0,255]) to produce integer-valued image data **759** in the allowed range.

If it is necessary to show the data before manipulation, for example when the image manipulation is an interactive process, the image can optionally be sent to a display **763**. The image is operated on by a manipulator **767** to produce some desired change. If the result of the manipulation is non-integer data then the image data may be passed to another integer converter and clipper **773**. In this way the image data **777** may again be displayed **763**. The new real-domain image data **777** are passed to the forward DCT-er **781** and the resulting DCT coefficients are input to the quantizer **785** to produce new quantized DCT coefficients **789**. These coefficients **789** are then input to the entropy encoder **793** to produce new coded data 'C1' **797** which are different from the original coded data 'C' **715**. Now the new coded data 'C1' **797** incorporates not only the desired changes made to the image by the image manipulator **767**, but also the errors resulting from the integer converter and clippers **755** and **773**.

Figure 8(a) shows the JPEG example luminance quantization matrix **804** for 8x8 DCT luminance blocks. Figure 8(b) gives the JPEG example chrominance quantization matrix **814** for 8x8 DCT chrominance blocks. The smallest quantization value in Figure 8(a) is 10. The smallest quantization value in Figure 8(b) is 17. Since the maximum possible error from rounding is 0.5 for each of 64 samples, the largest error in the unquantized forward transform coefficients from conversion to integers by rounding is 4 (shown in Figure 8(c)) for JPEG. For the quantization matrices shown in Figures 8(a) and 8(b) this size error is

less than half of all of the values and will disappear during quantization. However, for high quality applications such as high end printing or digital studio editing, the quantization matrix values are much smaller. In some cases, the DC (upper left corner) term is as small as 1 to preserve maximum quality. Then the rounding errors are significant.

The maximum possible error from truncating is just under 1 for each sample. This almost doubles the error in the unquantized forward transform coefficients. For the quantization matrix in Figure 8(a) eight quantization values are small enough for this error to potentially change the transform-coded data.

A numerical example showing the multi-generation problem is given in Figure 8(c). In this example the transform used is the 8×8 DCT as used in the JPEG still image compression standard. A set of transform-domain coefficients **822**, of which only one (the constant, or DC, term) is non-zero, are operated on by the inverse transform to produce an block of real-domain data **824**. In this case the data consist of 64 values which are all equal to 128.5. Note that the JPEG level shift of 128 for 8 bit data has been applied. The real-domain data are rounded to the nearest integers **826**, which in this case means that each value is rounded up to 129. The forward transform is then applied to produce new transform-domain coefficients **828**. It can be seen that the resulting new transform coefficients **828** are significantly different from the initial transform coefficients **822**. This is a highly undesirable result.

This example also applies to transform-coded data if the DC quantization value is set to 1, 2, or 4. Then the transform coefficients **822** would be produced from transform-coded values of 4, 2, or 1 respectively. The quantization of the new transform coefficients **828** would change the resulting DC quantization values to 2, 4, or 8 respectively.

Another numerical example showing the multi-generation problem is given in Figure 8(d). Again the transform used is the 8×8 DCT as used in the JPEG still image compression standard. A set of transform-domain coefficients **832**, of which only one (the constant, or DC, term) is non-zero, are operated on by the inverse transform to produce a block of real-domain data **834**. In this case the data consist of 64 values which are all equal to 128.875. Note that the JPEG level shift of 128 for 8 bit data has been applied. The real-domain data are truncated to the nearest integers **836**, which in this case means that each value is reduced to 128. The forward transform is then applied to produce new transform-domain coefficients **838**. It can be seen that the resulting new transform coefficients **838** are significantly different from the initial transform coefficients **832**. This is a highly undesirable result.

Having demonstrated the errors caused by real-domain rounding or truncating when converting to integers, we now show how real-domain clipping can cause errors. Figure 8(e) shows an example of real-domain clipping **850**. This example uses the one-dimensional DCT to illustrate the problem.

Figure 8(d) shows a bar chart **854** displaying one block of data consisting of eight samples. The data displayed has only two frequency components: a constant, or DC, component which is indicated by the dotted line; and an alternating, or AC, component which gives an alternating wave pattern symmetrical about the dotted line. The magnitudes of these components, namely the respective DCT coefficients, are high-precision numbers. When quantization is performed, these DCT coefficients are rounded to the nearest quantization level. The data after transform-domain quantization are shown in the bar chart **858**. In the example shown, the DC coefficient has a small quantization value and so quantization does not change the DC level significantly. The AC coefficient shown has a large quantization value and so is changed significantly by quantization. This example shows the AC component almost doubling in magnitude due to quantization. These quantization values reflect, for example, those used when compressing chrominance image data. Thus the data represented after quantization have parts which have negative values. This shows how transform-domain data which, after inverse transforming, give real-domain negative values can be produced by original real-domain data which do not contain negative values.

Bar chart **862** shows the data produced from that in chart **858** after real-domain clipping. Those negative parts of the real data have been changed to 0. This results in the DC coefficient of the data increasing and hence leads to error being introduced. Because the quantization value for the DC coefficient is generally small, the error is large

enough to cause a change in the quantized data as given in Equation (3).

To further illustrate the possibility of error introduced by real-domain clipping, a numerical example **870** is shown in Figures 8(f) and 8(g). This example employs the system illustrated in Figure 5. This example uses the two-dimensional 8×8 DCT as used for transform coding of images to illustrate the problem described above. The initial quantized DCT coefficients are shown in matrix **874**. All but two of the coefficients are 0; the two non-zero coefficients are the DC coefficient and one high-frequency coefficient. The coefficients, after dequantizing using the quantization matrix shown in Figure 8(a), are shown in matrix **878**. When the inverse DCT is performed on these transform data and the level shift of 128 added, real data are produced as shown in matrix **882**. The data shown in matrix **882** have already been rounded to integers but have not been clipped to an allowed range. It can be seen that these real data include several negative values. After clipping, the real data **882** produce clipped real data as shown in matrix **886**. These data are identical to **882** except that each negative value has been replaced by 0. The forward DCT is then applied to the real-domain data to give new rounded transform data **890**. It can be seen that the new transform data are significantly different from the previous transform data **878**. When quantization is performed using the quantization matrix shown in Figure 8(a), new transform-coded data **894** are produced. The resulting changes in the transform data are large enough to produce

changes in the transform-coded data after quantization. This is a highly undesirable result.

In many situations, the process of decoding, manipulation and re-encoding of data needs to be done multiple times. In these situations each iteration of this process is referred to as a "generation". The errors described above, caused by converting to integers and clipping to an allowed range in the real domain, accumulate as multiple iterations are performed and may result in significant degradation of the data. It is realized that the foregoing are only representative examples of errors introduced by rounding (or truncating) and/or clipping. Other examples having more or less error developed are possible.

The problem is usually even worse following multiple generations of decoding and re-encoding as shown in Figure 9. Initial transform-coded data 'D0' **910** is dequantized and inverse transformed **920**, converted to integers and clipped to an allowed range **930** to produce integer-valued real-domain data **940**. The real-domain data **940** are passed to the forward transform and quantized **950** to give new transform-coded data 'D1' **960**. This whole process is iterated several times, and after some number 'n' of iterations the final transform-coded data 'Dn' **970** is produced. Because of errors in each step the final data 'Dn' **970** may be very different from the original data.

A case showing the problem significantly worsened due to multiple generations of real-domain manipulation of transform-coded data is shown in Figure 10. In addition to

the steps shown in Figure 9, some form of manipulation **310** is performed on the real-domain data, followed by converting to integers and clipping **320**. After the forward transform and quantization, the resulting quantized transform coefficients **1010** contain some error as in Figure 5. After 'n' generations, the final transform quantized coefficients **1020** may have quite large undesired errors.

DETAILED DESCRIPTION OF THE INVENTION

An example embodiment of a method for processing transform data with reduced error **1100** is illustrated in Figure 11(a). Transform data 'A' **110** are passed through an inverse transform **120** to produce high-precision real-domain data **130**, as in Figure 1(a). If it is necessary to pass the real-domain data to an output device **160** which takes integer-valued input, or to generate integer-valued data before manipulation for any other reason, the steps of converting to integers and clipping to an allowed range **140** is done before manipulation without affecting the high-precision real-domain data. The desired manipulation **1110** of the real-domain data is performed using a method which accepts high-precision data as input and produces high-precision data **1120** as output. This manipulation method **1110** performs conceptually the same processing on the data as the manipulation on integers **310** described above in Figure 3, but operates instead on high-precision data. If it is necessary to pass the manipulated real-domain data to an output device **160** which takes integer-valued input, or to

generate integer-valued data after manipulation for any other reason, the steps of converting to integers and clipping to an allowed range **140** are done after manipulation without affecting the high precision of the processed data.

An example embodiment of a system for processing transform data with reduced error **1105** in accordance with the present invention is illustrated in Figure 11(b). Transform data 'A' **115** are passed through an inverse transformer **125** to produce high-precision real-domain data **135**, as in Figure 1(b). If it is necessary to pass the real-domain data to an output device **165** which takes integer-valued input, or to generate integer-valued data before manipulation for any other reason, the integer converter and clipper **145** operates before manipulation without affecting the high-precision real-domain data. The manipulator **1115** operates on the real-domain data accepting high-precision data as input and producing high-precision data **1125** as output. This manipulator **1115** performs conceptually the same processing on the data as the manipulation on integers **310** described above in Figure 3, but operates instead on high-precision data. If it is necessary to pass the manipulated real-domain data to an output device **165** which takes integer-valued input, or to generate integer-valued data after manipulation for any other reason, the integer converter and clipper **145** operates after manipulation without affecting the high precision of the processed data.

An example of an embodiment of the present invention employing a method for performing inverse transform followed

by forward transform steps **1200** is illustrated in Figure 12(a). Transform data 'A' **110** are passed through an inverse transform **120** to produce high-precision real-domain data **130**, as in Figure 1(a). If it is necessary to pass the real-domain data to an output device **160** which takes integer-valued input, or to generate integer-valued data for any other reason, the steps of converting to integers and clipping to an allowed range **140** are done without affecting the high-precision real-domain data. The high-precision data **130** are used as input to the forward transform **1210**, which accepts real-valued data as input. The resulting transform data 'A3' **1220** are identical to the original transform data 'A' **110** which were the input to the inverse transform **120** if the forward transform **1210** is the inverse of the inverse transform since the errors from rounding and clipping are not present in the transform data 'A3'. The forward transform **1210** will produce different transform data 'A3' **1220** when a different forward transform is used. This allows conversion between transforms without the errors from rounding and clipping being present in the forward transform input.

An example of an embodiment of the present invention employing a system with an inverse transformer followed by forward transformer **1205** is illustrated in Figure 12(b). Transform data 'A' **115** are passed through an inverse transformer **125** to produce high-precision real-domain data **135**, as in Figure 1(b). If it is necessary to pass the real-domain data to an output device **165** which takes integer-valued input, or to generate integer-valued data for

any other reason, the integer converter and clipper **145** operates without affecting the high-precision real-domain data **135**. The high-precision data **135** are used as input to the forward transform **1215**, which accepts real-valued data as input. The resulting transform data 'A3' **1225** are identical to the original transform data 'A' **115** which were the input to the inverse transformer **125** if the forward transformer **1215** implements the inverse of the inverse transform since the errors from rounding and clipping are not present in the transform data 'A3'. The forward transformer **1215** will produce different transform data 'A3' **1225** when a different forward transformer is used.

Figure 13(a) shows a method for performing real-domain manipulation of transform data with reduced error **1300**. This method is formed by extending the method **1100** described in Figure 11(a). In this case, the high-precision data **1120** are passed as input to a forward transform **1210** which accepts high-precision data as input, to produce new transform data 'A4' **1310** without rounding and/or clipping errors.

Figure 13(b) shows a system for performing real-domain manipulation of transform data with reduced error **1305**. This method is formed by extending the system **1105** described in Figure 11(b). In this case, the high-precision data **1125** are passed as input to a forward transformer **1215** which accepts high-precision data as input, to produce new transform data 'A4' **1315** without rounding and/or clipping errors.

A method for performing real-domain manipulation of transform-coded data with reduced error is illustrated in

Figure 14(a). Figure 14(a) shows integer transform-coded data 'B' **210** are dequantized **220** and the output passed through an inverse transform **120** to produce high-precision real-domain data **130**, as in Figure 2(a). If it is necessary to pass the real-domain data **130** to an output device **160** which takes integer-valued input, or to generate integer-valued data **150** before manipulation for any other reason, the steps of converting to integers and clipping to an allowed range **140** are done before manipulation without affecting the high-precision real-domain data **130**. The desired manipulation **1110** of the real-domain data is then performed using a method which accepts high-precision data as input and produces high-precision data **1410** as output. This manipulation **1110** performs conceptually the same processing on the data as the manipulation on integers **310** described above in Figure 3, but operates instead on high-precision data. If it is necessary to pass the manipulated real-domain data to an output device **160** which takes integer-valued input, or to generate integer-valued data after manipulation for any other reason, the steps of converting to integers and clipping to an allowed range **140** are done after manipulation **1110** without affecting the high precision of the processed data **1410**.

A system for performing real-domain manipulation of transform-coded data with reduced error is illustrated in Figure 14(b). Figure 14(b) shows integer transform-coded data 'B' **215** input to an inverse quantizer **225** and passed through an inverse transformer **125** to produce high-precision real-domain data **135**, as in Figure 2(b). If it is necessary

to pass the real-domain data **135** to an output device **165** which takes integer-valued input, or to generate integer-valued data **155** before manipulation for any other reason, the integer converter and clipper **145** operates on the data before manipulation without affecting the high-precision real-domain data **135**. The desired manipulation of the real-domain data is then performed using a manipulator **1115** which accepts high-precision data as input and produces high-precision data **1415** as output. This manipulator **1115** performs conceptually the same processing on the data as the manipulation on integers **310** described above in Figure 3, but operates instead on high-precision data. If it is necessary to pass the manipulated real-domain data to an output device **165** which takes integer-valued input, or to generate integer-valued data after manipulation for any other reason, the integer converter and clipper **145** operates on the non-integer data **1415** after manipulation **1115** without affecting the high precision of the processed data **1415**.

An example embodiment of a method for real-domain conversion of transform-coded data **1500** is shown in Figure 15(a). The high-precision data **130** are used as input to the forward transform **1210**, which accepts real-valued data as input. The output of the forward transform **1210** is quantized **1510**. Depending upon the desired system implementation, the forward transform operation **1210** may employ a different transform than that used in the inverse transform operation **120**. For example, the inverse transform **120** may use the inverse DCT transform whereas the forward transform **1210** may

use the Fourier transform. The resulting integer transform-coded data 'B2' **1520** are identical to the original integer transform-coded data 'B' **210** which were the input to the inverse quantize step **220** if the forward transform operation **1210** is the inverse of the inverse transform operation **120** and the quantization values used in the inverse quantization step **220** and the quantization step **1510** are identical. It is noted that the forward transform **1210** will produce different integer transform-coded data 'B2' when a different forward transform is used. Similarly, use of different quantization values in the inverse quantization **220** and quantization **1510** also produces different integer transform-coded data **1520**. This method thus allows conversion between transforms and quantization matrices without the errors from rounding and clipping being present in the forward transform **1210** input **130**.

The conversion between quantization matrices may be for coarser or finer quantization. For converting data from the JPEG international standard to the MPEG international standard, the quantization is likely to be coarser. The higher quality JPEG independent images are needed during the editing process. The coarser, more compressible, MPEG images are used to achieve the desired bandwidth objectives. On the other hand, in recompressing JPEG images after significant hand editing, the quantization is likely to be finer in order to preserve the changes.

An example embodiment of a system for real-domain conversion of transform-coded data **1505** in accordance with the present invention is shown in Figure 15(b). The

high-precision data **135** are used as input to the forward transformer **1215**, which accepts real-valued data as input. The output of the forward transformer **1215** is input to the quantizer **1515**. Depending upon the desired system implementation, the forward transformer **1215** may produce a different transform than that used in the inverse transformer **125**. For example, the inverse transformer **125** may use the inverse DCT transform whereas the forward transformer **1215** may use the Fourier transform. The resulting integer transform-coded data 'B2' **1525** are identical to the original integer transform-coded data 'B' **215** which was the input to the inverse quantizer **225** if the forward transformer **1215** produces the inverse of the inverse transformer **125** and the quantization values used in the inverse quantizer **225** and the quantizer **1515** are identical. It is noted that the forward transformer **1215** will produce different integer transform-coded data 'B2' when a different forward transform is produced. Similarly, use of different quantization values in the inverse quantizer **225** and quantizer **1515** also produces different integer transform-coded data **1525**. This system thus allows conversion between transforms and quantization matrices without the errors from rounding and clipping being present in the forward transformer **1215** input **135**.

A method for performing real-domain manipulation of transform-coded data with reduced error **1600** is formed by extending the method **1400** described in Figure 14(a) as is illustrated in Figure 16(a). The high-precision data **1410** are passed as input to a forward transform **1210** which

accepts high-precision data as input. The output values from the forward transform are quantized **1510** to produce new transform-coded data 'B3' **1610**.

A system for performing real-domain manipulation of transform-coded data with reduced error **1605** is formed by extending the method **1405** described in Figure 14(b) as is illustrated in Figure 16(b). The high-precision data **1415** are passed as input to a forward transformer **1215** which accepts high-precision data as input. The output values from the forward transformer are input to the quantizer **1515** to produce new transform-coded data 'B3' **1615**.

An example embodiment of a method for real-domain manipulation of transform-coded data with reduced error **1700** is shown in Figure 17(a). The chosen embodiment is a method for real-domain manipulation of coded images, which are transform-coded using the DCT. Coded data 'C' **710** are decoded by a lossless entropy decode step **720** to produce quantized DCT coefficients. These coefficients are dequantized **730** and passed through an inverse DCT **740** to produce high-precision real-domain data **1710**. If it is necessary to pass the image before manipulation to a display device **758** which takes integer-valued input, or to produce integer-valued data **754** before manipulation for any other reason, the steps of converting to integers and clipping to an allowed range **750** are performed before manipulation **1720** without affecting the high-precision real-domain image data **1710**. The desired manipulation **1720** of the image is then performed using a method which accepts high-precision data as input and produces high-precision data **1730** as output.

If it is necessary to pass the manipulated image data to a display **758** which takes integer-valued input, or to generate integer-valued image data **1750** after manipulation for any other reason, the steps of converting to integers and clipping to an allowed range **1740** are performed after manipulation **1720** without affecting the high precision of the processed image data **1730**. The high-precision image data **1730** are passed as input to a forward DCT **1760** which accepts high-precision data as input. The output values from the forward transform **1760** are quantized **780** to produce new integer DCT coefficients **1770**. These coefficients **1770** are encoded by a lossless entropy encode step **788** to produce new coded data 'C2' **1780**. If the forward and inverse transforms and the manipulation system are sufficiently accurate so that the error they introduce is less than half a quantization step, as described in Equation (3) given above, no error at all is introduced to the DCT coefficients.

An example invention embodiment of a system for real-domain manipulation of transform-coded data with reduced error **1705** is shown in Figure 17(b). The chosen embodiment is to implement a method for real-domain manipulation of coded images such as JPEG-coded images, which are transform-coded using the DCT. Coded data 'C' **715** are decoded by a lossless entropy decoder **725** to produce quantized DCT coefficients. These coefficients are sent to a inverse quantizer **735** and then passed through an inverse DCT-er **745** to produce high-precision real-domain data **1715**. If it is necessary to pass the image before manipulation to

a display device **763** which takes integer-valued input, or to produce integer-valued data **759** before manipulation for any other reason, the integer converter and clipper **755** produces integer-valued data in the allowed range before manipulation **1725** without affecting the high-precision real-domain image data **1715**. The manipulator **1725** which performs the desired manipulation of the image accepts high-precision data as input and produces high-precision data **1735** as output. If it is necessary to pass the manipulated image data to a display **763** which takes integer-valued input, or to generate integer-valued image data **1755** after manipulation for any other reason, the optional integer converter and clipper **1745** produces integer-valued data **1755** after the operation of the manipulator **1725** without affecting the high precision of the processed image data **1735**. The high-precision image data **1735** are passed as input to a forward DCT-er **1765** which accepts high-precision data as input. The output values from the forward DCT-er **1765** are sent to the quantizer **785** to produce new integer DCT coefficients **1775**. These coefficients **1775** are encoded by a lossless entropy encoder **793** to produce new coded data 'C2' **1785**. If the forward and inverse transforms and the manipulation system are sufficiently accurate so that the error they introduce for each coefficient is less than half a quantization step, as described in Equation (3) given above, no additional error is introduced to the DCT coefficients.

A method for performing real-domain manipulations of transform-coded data with reduced error in multiple steps **1800**, alternating the manipulation steps with forward

transforming and quantizing steps and inverse transform and quantizing steps, is illustrated in Figure 18(a). In general each manipulation may perform another operation on the data. For example for digital studio editing, the first manipulation might color correct the image. The second manipulation might merge the color corrected image with a background using the chroma-keying method. The third manipulation might add highlights to the image. The fourth manipulation might crop the image to convert from the 16:9 width to height aspect ratio of movies to the 4:3 aspect ratio of television. For the printing of images the first manipulation might rotate the image 90 degrees to orient the image with the printing direction. The second manipulation might merge several independent images into one composite image. A third manipulation might do a color conversion.

As shown in Figure 18(a) transform-coded data 'D0' **910** are dequantized and passed through an inverse transform **920** to produce high-precision real-domain data **1810**. If it is necessary to produce integer-valued data for any reason, the high-precision data **1810** may be converted to integers and clipped to an allowed range **1820** without affecting the high precision of the real-domain data **1810**. The desired manipulation **1110** of the real-domain data is then performed using a method which accepts high-precision data **1810** as input and produces high-precision data **1840** as output. If it is desired to produce an integer-valued of this output data, the high-precision data **1810** may be converted to integers and clipped to an allowed range **1830** without affecting the high precision of the output data. The

high-precision output data are passed as input to a forward transformer and quantizer **1850** to produce new transform-coded data 'F1' **1860**. The process of inverse quantizing and inverse transforming, manipulation and forward transforming and quantizing may be repeated multiple times with the manipulation **1870** being different upon each iteration. After multiple steps, final transform-coded data 'Fn' **1880** are produced with rounding and/or clipping errors reduced or eliminated. Outputs resulting from any of the convert to integer and clip steps may be sent to an output device **1890** with or without a multiplexor.

An example invention embodiment of a system for performing real-domain manipulations of transform-coded data with reduced error in multiple stages **1805**, alternating the operation of a manipulator with the operation of a forward transformer and quantizer and the operation of an inverse quantizer and inverse transformer, is illustrated in Figure 18(b). Transform-coded data 'D0' **1815** are fed to an inverse quantizer and inverse transformer **1819** to produce high-precision real-domain data **1823**. If it is necessary to produce integer-valued data for any reason, the high-precision data **1823** may be operated on by the integer converter and clipper **1827** without affecting the high precision of the real-domain data **1823**. The manipulator **1115** then operates on the real-domain data **1823** to produce the desired manipulation and produces high-precision data **1845** as output. If it is desired to produce integer-values of this output data, the high-precision data **1845** may be input to an integer converter and clipper **1835** without

affecting the high precision of the output data. The high-precision output data are passed as input to a forward transformer and quantizer **1855** to produce new transform-coded data 'F1' **1865**. The steps of inverse quantizing and inverse transforming, manipulation and forward transforming and quantizing may be repeated multiple times with the manipulator **1875** being different upon each iteration. After multiple iterations, final transform-coded data 'Fn' **1885** are produced with real-domain rounding and/or clipping errors reduced or eliminated. In a particular embodiment the output from any or all of the integer converter and clipper modules is fed to the output device **1895**. For coded image data the output device may be a display or television set. For coded audio data the output device may be a player and/or recorder.

A numerical example showing how the present invention solves one aspect of the multi-generation problem is given in Figure 19(a). A set of transform-domain coefficients **822**, of which only one (the constant, or DC, term) is non-zero, are operated on by the inverse transform to produce an block of real-domain data **824**. In this case the data consist of 64 values which are all equal to 128.5. Note that the JPEG level shift of 128 for 8 bit data has been applied. The forward transform is then applied to produce new transform-domain coefficients **1910**. It can be seen that the new transform coefficients **1910** are identical to the initial transform coefficients **822**. This is due to the rounding error not being present in the data sent to the forward DCT.

Another numerical example showing how the present invention solves another aspect of the multi-generation problem is given in Figure 19(b). A set of transform-domain coefficients **832**, of which only one (the constant, or DC, term) is non-zero, are operated on by the inverse transform to produce an block of real-domain data **834**. In this case the data consist of 64 values which are all equal to 128.875. Note that the JPEG level shift of 128 for 8 bit data has been applied. The forward transform is then applied to produce new transform-domain coefficients **1938**. It can be seen that the new transform coefficients **1938** are identical to the initial transform coefficients **832**. This is due to the truncation error not being present in the data sent to the forward DCT.

Having demonstrated how using the high-precision numbers removes the errors caused by real-domain rounding or truncating, we now show how real-domain clipping errors are also avoided. The same numerical starting point and first three steps used in Figure 8(f) are shown in Figure 19(c). The initial quantized DCT coefficients are shown in matrix **874**. All but two of the coefficients are 0; the two non-zero coefficients are the DC coefficient and one high-frequency coefficient. The coefficients after dequantizing are shown in matrix **878**. The quantization matrix used is shown in Figure 8(a). When the inverse DCT is performed on these transform data, real data are produced as shown in matrix **882**. The data shown in matrix **882** have already been rounded to integers but have not been clipped to an allowed range.

Figure 19(d) shows the results of the forward DCT applied to the real-domain data to give new rounded transform data **1944**. When quantization is performed, new transform-coded data **1948** are produced. In this example, the changes in the transform data are not large enough to produce changes in the transform-coded data after quantization.

Examples of the manipulation between generations include merging two or more transform-coded data sets. For transform-coded image data sets, the merging may be needed because multiple small images need to be collected into one bigger picture. Fan-folded advertising brochures typically are composed of multiple individual pictures. Today's highest end laser printers print more than one page at a time. In such cases, the images generally do not overlap, but may not have the same quantization, positioning relative to the reference grid such as the 8x8 block structure for JPEG DCTs, or orientation. By composing the final picture in the real domain, standard processes can be used for each subimage. Then the composite image can be re-compressed for eventual decompression for on-the-fly printing.

Similarly, digital editing can include many special effects requiring several independent manipulations performed serially. Digital movies often use the fade-in/fade-out special effect to perform a smooth transition between two key scenes. Such special effects may follow independent processing of each scene. Thus, multiple generations of decompression and recompression are often needed in the editing to produce the composite of the special effects.

Chroma-keying involves two independent video data streams. In one video stream the background has been captured. In the other video stream the foreground, often composed of action involving live actors, has been filmed against a blank single color such as a deep blue or black background. Then the blank pixels in the foreground image are replaced with pixels from the background video. Since the pixels are being mixed at a single-pixel level, the images need to be combined in the real domain. The errors introduced by converting to integers and clipping are highly undesirable for such digital studio applications. These errors are reduced or eliminated by implementing the present invention.

Another application example for use of the present invention is in the high-end digital graphics market which uses digital images with sometimes more than 100 megapixels. Glossy advertising brochures and the large photographic trade show booth backdrops are just two examples of the use of such high quality digital imagery. High-quality lossy JPEG compression are sometimes used to keep the transmission and storage costs down. As such images are decompressed and recompressed to allow changes and modifications such as adding highlights, correcting colors, adding or changing text and image cropping, unintentional changes are a problem that is solved with the use of the concepts of the present invention.

The above examples for the concepts of the present invention are usual for image and video transform data. The wide use of the Internet has shown the value of JPEG and MPEG compressed image data. When JPEG images are to be

printed, then manipulations such as a change of scale or a change of orientation may be required. In addition, a transformation to another color space followed by recompression will allow the print-ready versions of the image to be stored. Use of the present invention overcomes the problem inherent in propagating the errors from the rounding and clipping.

Audio coded data also needs to be decompressed, mixed with special sound effects, merged with other audio data, edited and processed in the real domain with reduced errors. Similar implementations are performed for other industrial, commercial, and military applications of digital processing employing a transform and an inverse transform of data representing a phenomenon when the data is stored in the transform domain. These are thus other representative applications wherein use of the present invention is highly advantageous.

It is further noted that this invention may also be provided as an apparatus or a computer product. For example, it may be implemented as an article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for causing a computer to perform the methods of the present invention.

It is noted that although the description of the invention is made for particular arrangements of steps, the intent and concept of the present invention are suitable and applicable to other arrangements. It will be clear to those skilled in the art that other modifications to the disclosed embodiments can be effected without departing from the

spirit and scope of the invention.